並列計算機PACSによる数値シミュレーション (数値計算のアルゴリズムの研究)

AUTHOR(S):

星野, 力

CITATION:

星野, 力. 並列計算機PACSによる数値シミュレーション (数値計算のアルゴリズムの研究). 数理解析研究所講究録 1982, 453: 19-41

ISSUE DATE:

1982-02

URL:

http://hdl.handle.net/2433/102994

数理解析研究所講究録
453巻 1982年 19-41

並列計算機PACSによる数値シミュレーション

筑波大 構造　　星野　力

ノイマン型汎用計算機上で最適であったアルゴリズムが、非ノイマン型にとっても そうであるとは限らない。アルゴリズムの適不適は、計算機アーキテクチャと具体的な応用問題に依存している。ここでは試作の完了した科学技術計算専用の並列計算機PACS-32により種々の応用問題をといた結果と、これからの数値計算アルゴリズムの見通しについてのべる。

アルゴリズムの良し悪しを論ずる前に

ユーザにとっては自明の（計算機やアルゴリズムの研究者にとっては必ずしも自明でない？）いくつかの前提を述べておく。

1. ユーザのもつ巨大科学技術計算のニーズは現在の計算機能力より桁違いに大きい。今の計算機より1万倍高速の

約10^{10}FLOPS（浮動小数点演算／秒）の速度が要求されている。10倍ぐらいの速度向上などでは焼け石に水である。

2. 白い猫であろうが黒い猫であろうが，ねずみをとる猫がよい猫であるように，ユーザにとって良いコンピュータ，良いアルゴリズムとは問題を高速度に解くものがそれである。コンピュータ・アーキテクチャが独創的でなくても，アルゴリズムが最近はやらないものであっても一向にかまわない。

3. もちろんコンピュータもアルゴリズムも使いやすい方がよいに決っている。しかし使いにくいのを克復して1万倍に高速化した計算結果と，使いやすいので10倍にとどまった計算結果とでは勝敗は自明である。学会で両方の発表が並んでいる様を想像してほしい。

4. ユーザにとって計算機もアルゴリズムも，特定の応用問題を解くためにあり，そのためにだけ存在する。あるアルゴリズムや計算機にとって常に意地悪い例題は存在するが，そういう例題が現実の応用問題として多く存在するかどうかが問題なのである。

5. ユーザにとって大型汎用計算機は必要でもなく経済的でもなくなってきている。汎用機は科学計算も事務計算も何でもやれるのが値打ちである。何でもやれる機種は

本社に一台しか置けない時代では最適であったが、ユーザごとに分散処理する時代ではそうではない。科学計算しかやらないユーザにとって100%の汎用性があっても要らないものを買わされていると思うだけである。

3A間のからみ

ノイマン型汎用機は（計算可能性のあるものならば）何でも出来たので、アーキテクチャのことは一応わすれて、アプリケーションに適したアルゴリズムを考えてゆけばよかった。

しかし計算機のアーキテクチャが各アプリケーションごとに専用化してゆくと，アルゴリズムもアプリケーションとアーキテクチャを十分意識して考えねばならない。この3つのAの間には汎用機時代では存在しなかったからみが生じてくる。

世の中には色々な人がいて、面白いアーキテクチャを考えてから、これは何に使えそうかと考える人もいるし、またアルゴリズムの研究に凝っていて特定のアルゴリズムに最も適したアーキテクチャを考える人もいる。

しかしユーザにとっては，まず彼自身の問題があり、これを経済的制約の内で解くための最適のアーキテクチャとアルゴリズムは、何かというのが課題なのである。

22

ユーザのもつモデル

ではユーザのかんがえている代表的問題とは何か？　数学的表現に応じて分類してみよう

1. 連続体・場モデル

流体や各種の場のモデルで、数学的には偏微分方程式で記述されるのが普通である。場を形成する作用は「近接的」である。すなわち空間的に近接している媒質や場との相互作用により、ある点での媒質・場の物理的諸量が決る。

2. 粒子モデル

粒子の数が比較的少くなると、個々の粒子、或いは粒子の集団の運動の効果が支配的になる。粒子のうける力は粒子集団全体の作る場から受けたり、また系外から外力として与えられる。粒子のうける力は（少くとも表現方法に関しては）近接的とは限らない。数学的に表現すれば連立常微分方程式や偏微分方程式になる。

3. 構造・システムモデル

数学的にはベクトルやマトリックスが現われるモデルで構造物やシステムの定常状態，安定問題，固有値問題，数理計画法，信号処理などで，近接的でないのが特徴といえる．

作用の空間からプロセッサの空間への写像

物理現象の基本的メカニズムは近接作用であることは物理学の教えるところであるが、数学的表現は必ずしもそうとは限らなく、ましてや計算機中の計算の流れは、近接作用とは全く関係がなくなってくる。

しかし多数のプロセッサ（PU：Processing Unit）から構成される並列マルチプロセッサ・システムでは、元の問題での作用の流れ方からプロセッサの空間への写像が問題となってくる。

この写像が間接的になっている一例としてFMP（Flow Model Processor：バローズ社のNASA数値風洞プロジェクトNASFへの提案）を挙げる。(1) 図1参照。これは512台のPUが521バンクのメモリへ、ω-Networkといわれる Shuffle-Exchange Network を通じて結合されているものである。もとの物理空間では隣接格子点間に流れている作用は、このプロセッサ・メモリ空間上では、ずたずたにちぎれて取扱われている。

直接的な写像の例はILLIAC-IV(2)や、後で紹介するPACS(3)(4)である。図2参照。ここでは物理空間をそのままプロセッサ空間に網をかぶせるように写像する。近接作用はそのまま、隣接PU間の通信として扱われる。非近接的なモデルは、デー

タを隣接PU間で次々に渡してゆく方法（ルーティング）により行うか，制御ユニット（CU；Control Unit）経由で逐次的に行う。

直接的写像の方が計算機ハードウェアも作りやすいし，物理的イメージも保たれていて判りやすいのに何故間接的な写像をあえて採るかといえば100％の汎用性（すなわち任意のPU間で通信出来ること）を保とうと考えるからである。（直接的写像でもPU間で通信出来ることには変りない。問題はその効率である。）

しかしハードウェアが，どんどん安くなってきている時代に何故100％汎用の固定観念にとらわれつづけるのであろうか？　正確に云えば，LSI化のメリットの大きい論理部分が安くなっているのであってプロセッサ間結合を行う結線，ボードやコネクターの部分は必ずしもそうではない。これは複雑なアーキテクチャを考える人がしばしば忘れている点である。

またこれら結線部分は計算速度の低下をまねく。ジョゼフソン素子がいかに高速にスイッチングを行っても，ジョゼフソンコンピュータは決して高速にならない。せいぜいシリコンコンピュータの数10倍にすぎない。

超高速をねらうコンピュータの通信部分（プロセッサ間結合を行うネットワークとか，バッファメモリ）は極小にしなけ

ければならない。この点、最近のテキスト[5]の主張と全く同意見である。

PACS－32システムの概略

PACS（Processor Array for Continuum Simulation）については若干の報告[3][4]もあるので詳しくはそれを参照していただくことにし，ここでは数値アルゴリズムが関係する範囲内で概略を紹介する。

図3のように，8×4の2次元配列をしている32台のPU（1台のPUは2個のマイクロプロセッサと18Kバイトのメモリより成る）が，これを制御しているCU（同じタイプのマイクロコンピュータ）を介してHOST（汎用計算機ミニコンTI990/20）と結合されている。

32台のPUは，すべて独立のプログラムをもつコンピュータであり，上下左右方向の隣接PUとは，隣接PUとのみ共有するメモリCMを経由してデータをやりとりする。

PUは独立だから，その実行は同期する必要はない。データはCMに書いておけばとなりのPUがそれを必要とした時点で読み計算に用いる。

CU↔PU間の制御はステータスレジスタSRとコントロールレジスタCRとで行う。これはPUの同期や，収束判定，ジョブ終

などに用いられる。

並列計算の進み方に最終責任をもつのは，すべてユーザである。従来のFORTRANプログラムのままでOSが何もかも面倒を見てくれるというような「甘えの構造」にはなっていない。

もちろんユーザが並列プログラムを記述しやすいような言語（高水準言語SPLM，アセンブル言語MASP）が用意され，まだ言語化するに至っていない種々のユーティリティがある。

PACSのプロセッサの並べ方はILLIAC-IVに似ているので何か独創性が少いように感じる向きがあるようであるが，まずユーザにとってアーキテクチャの独創性などどうでもよいということを前に述べた。

しかし，それはさておきILLIAC-IVはSIMDであり，PACSはMIMDであることや，制御方式，レジスタやメモリの構成など全然似ていないと言ってもよい。この差は数値シミュレーション技術に大きな違いをもたらす。

したがってILLIAC-IVでこの種のアレイの可能性がすべて尽されたとは考えにくい。とくに近接性を本当に生かすアーキテクチャはMIMDであることを考えるならば，直角格子型アレイのもつ可能性をもっと調べる必要があると思われる。

PACSの並列処理モード

PACSでは大別して2つの処理モードが可能である。

1. 同期モードあるいは同期可能モード

プログラム上のある一点で，すべてのPUの実行をそろえるものである。必要に応じてCALL SYNC (CODE)とすれば同期がとれる。ただしCODEは同期コードである。

2. 非同期モード

CALL SYNC とやらなければ，すべて非同期といえる。

しかし隣接PUとのデータのやりとりに関して次の三種類がある。

(a) データフロモード

隣接PUのデータが新しいものにそろえば，自分の計算が開始出来る場合CM上にデータレディフラグを設けて通信する。偏微分方程式の反復法などでこのモードが使える。

(b) 完全非同期モード

隣接PUのデータの新旧にかゝわらず読み書きをする。ただし読み書きが競合しないように注意する。

(c) 非干渉モード

隣接PUのデータのやりとりが一切ないもの。試行の独立なモンテカルロ法がこれに当る。

PACSの応用

今までに解いた応用問題，とくにアルゴリズムについて述べる。

1. ポアソン方程式

PACSのアーキテクチャからすぐ反復法が最適のアルゴリズムであることがわかる。並列化効率のよいのは Odd-Even SOR である。すなわち格子点 (i,j) を $i+j=$ Odd の組と，$i+j=$ even の組にわけ，各組の格子点は同時に全PUで計算し，Odd と even の組を交互に計算するものである。

for $i+j=$ odd, $\quad 1<\omega<2$

$$\phi_{ij}^{(n+1)} = (1-\omega)\phi_{ij}^{(n)} + \omega\Big(\sum_{\substack{kl\in \\ \text{上下左右}}} \phi_{kl}^{(n)} + S_{ij}\Big)/4$$

for $i+j=$ even, $\quad 1<\omega<2$

$$\phi_{ij}^{(n+1)} = (1-\omega)\phi_{ij}^{(n)} + \omega\Big(\sum_{\substack{kl\in \\ \text{上下左右}}} \phi_{kl}^{(n+1)} + S_{ij}\Big)/4$$

計算テクニックとしては1つのPUには必ず偶数個の格子点を割りあてる。

並列処理モードは同期モードとし，① まず同期をとって

②隣接PUへデータを渡す。③その後は各PUごとに計算を行う。④次に同期をとり⑤STATUS変数（SR/CRレジスタに対応）を用いて行う。

計算はNXN正方領域周期境界条件の場合について行い次の結果をえた。くり返し回数は理論通りNに比例し，$T = 0.18\ N^3\ msec$ と書ける。ω は最適値 ω_{opt} にした。

N	10^{-5}までのくり返し回数	T：PACS時間	HOST時間
16	30	0.93 sec	1.41 sec
32	58	5.94	10.65
48	86	18.68	34.22

2. 沸騰水型原子炉炉心（3次元）における核・熱・水力的方程式のシミュレーション

今までにPACSで行った最大の計算である。図4にプログラムの構成をPAD表現(7)で示す。（もうそろそろフローチャートなどという生産性の低いものは廃棄した方がよい）

右端のブロックの中は変係数の連立（二元）ヘルムホルツ方程式（中性子拡散方程式）の固有値問題になっており，固有値ループの中にポアソン方程式のループを含んでいる。右端のループは，求まった最低固有値に対応する解自身の非線型関数として，ヘルムホルツ方程式の係数(核

断面積）を変更するループ，その外側（左側）のループは流量を制御して炉心を臨界状態にもってゆくループ，その外側は時間的な炉心の変化をゼノンの蓄積をおいかけてゆくループ，その外側は解をHOST計算機へ送って出力するループである。

ここでは，すべての式を掲げることは出来ないので文献(4)(8)を参照していただきたい，

3次元の炉心は図5のように2次元に投影して計算する。したがって1次元方向は1つのPUにより逐次的に計算される。

PACSのハードウェアの計算力は1PU当り0.0156 MFLOPS（10^6浮動小数点演算／秒）であり，32台では$0.0156 \times 32 = 0.5$ MFLOPSである。しかし，すべてのPUを有効に働かせることは出来ずオーバヘッド（計算力の無駄）が生じる。このためPU効率α＝有効計算／全計算と定義すると，PACSは32台で0.5α MFLOPSとなる。

並列処理はすべて同期モードで行われた。同期モードにおけるオーバヘッドとして考えられるものは2種類ある。

(1) 各PUごとに有効な計算量が異るため，同期点の直前で早く到着したPUはアイドル状態に入る。またはじめ

からそもそも計算をやってない炉心外領域のPUの計算は、すべてアイドルと考える。β_1＝アイドル／全計算＝0.219となった。

(ii) 有効な計算の内でも、隣接PUへデータを渡すための仕事は比較対象の普通の計算機ではしなくてもよい仕事だから、これもオーバヘッドと考える。β_2＝隣接通信／有効計算＝0.035となった。

よって$\alpha=(1-\beta_1)(1-\beta_2)=0.754$となった。

すなわち32台のうち24台分が有効に働いていると言える。速度比較は次の通り。

機種	CPU時間	名目速度	実効速度
PACS-32	3986 sec	0.5 MFLOPS	0.377 MFLOPS
M200	340.9 sec	3.3 MFLOPS	

はじめから仕事を割りあてられていない炉心外PUを除いて考えれば（すなわち炉心内にのみ限定して考えれば）$\alpha=0.899$となる。

重要なことは　このPU効率αはPU台数に無関係なことである。この汎用を意図したCm^*のようなシステムがPU 10台ぐらいでαが急減するのとは根本的に

異る．PACS はもし 10000 台の PU で（200×200 格子点の）同様の計算をやったとすれば，7540 倍に速度向上するであろう。

なお「並列プロセッサ・システムで実用的な計算の出来た例がまだない」とよくいろいろな記事に書かれているが，この炉心計算プログラムは簡単なものとは言え原子力発電所に実装しても十分使える程，実用的なものであることを指適しておく。

3. 一次元気体力学方程式

衝撃波の伝播を次の方程式をといて求める。(1)

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) = 0 \qquad \text{----質量保存式}$$

$$\frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}(p + \rho u^2) = 0 \qquad \text{----運動量保存式}$$

$$\frac{\partial}{\partial t}\left\{\rho\left(e + \frac{u^2}{2}\right)\right\} + \frac{\partial}{\partial x}\left\{\rho u\left(e + \frac{u^2}{2} + \frac{p}{\rho}\right)\right\} = 0 \quad \text{--- エネルギー保存式}$$

$$p = (\gamma - 1)\rho e \qquad \text{-------状態方程式}$$

これは，1 次元の計算も PACS の 2 次元プロセッサアレイ上で可能であることを示すためのものである。PACS は 4 方向の PU との通信が可能であるが，このうち 2 方向

を使えば1次元の計算が出来る。計算時間の例を示す。

250時間ステップ，512空間格子点，同期モードで処理

スキーム	PACS-32	TI990/20
Friedrichs-Lax	31.27 sec	144 sec
Explicit-Euler	34.48	161
Lax-Wendroff	43.95	227
Godunov	26.94	128

4. 常微分方程式

簡単なものであるが

$$\dot{x} = -y$$

$$\dot{y} = \min(a^2 x, b)$$

を同期モードでとく。計算時間は1600ステップで PACS 3秒，しかしデータのHOSTへのとり出し，出力などのため，全体では30秒かかる。

5. プラズマ中の高エネルギー粒子のモンテカルロ計算[10]

まず各PUは粒子の初速度をCUより受けとる。軌道を記述する常微分方程式を予測子修正子法で積分し，粒子が体系外へ逃れるか，または，あるエネルギー以下になると計算を終り結果をCUへ返す。以下これをくり返す。

粒子相互は干渉しないとしてあるモデルなので，各PUの計算は無関係となり非同期非干渉モードで処理した。

計算機の技術としては通常のマルチプロセッサと同じで，特に難しい問題はない。

たゞHOST ↔ CU ↔ PU間データの転送速度が低いとPUの待ちが生じ，全体の効率はPU台数とともに低下する。この例では32台PUで$\alpha = 0.76$となっている。

えられた速度はPACS 817秒で，M200（117秒）の約1/7の速度である。

これからの問題；並列特有のアルゴリズム

(i) 非同期反復法

普通の反復法では，すべての格子点は同一回の反復をうけるが，非同期モードでは一般に反復回数は格子点ごとに異ってくる。隣接PUの旧いデータを使って反復する欠点と，反復回数が部分的に増える長所があり，計算負荷に不均一がある場合，非同期方式が有利になると言われている。(12)

(ii) 粒子シミュレーション(11)

粒子と場を同時に並列プロセッサアレイで扱うには，両者のデータの表現方式について考えねばならない。

出来るだけPU利用効率を上げるには，計算負荷を均等化するべきである。

粒子のデータ表現にはラグランジュ表現が適している。

すなわち各PUに同数の粒子を割りあて粒子の座標や速度などのデータを格納する。

一方場の表現にはオイラー表現が適している。すなわち同体積の各部分領域に分割し，場のデータをPUに分担させる。

場の計算と粒子の運動の計算は交互にくるから両データ表現間の変換をうまくやらないといけない。また粒子はどこかへ集まったりもする。すべて直達力（万有引力のように）として積分形式にすることも考えられる。PU負荷を均一にする表現でしかも近接性も損わないような方式については，これから研究の必要なテーマである。

これらは，このような並列処理特有のアルゴリズムの研究もどしどし行われてゆくであろう。

おわりに

土俵が変れば相撲のとり方も変るように，計算機が変ればアルゴリズムも当然変るものである。しかし計算機は独り歩きするものではなくてユーザのもっているアプリケーションが独立変数なのである。ここでは並列プロセッサ型のPACS-32による並列計算の実例を紹介した。

PACSは数百台のPUをもつシステムへ拡張することを計画しており，そのPUをVLSI化することにより，数万台のPUをもつ約10GFLOPSのVLSIプロセッサアレイを作ることを最終目標としている。これを80年代に実現することは，急速に進歩しているLSI技術を前提に考えれば，それほど難しいことではないであろう。

VLSI化したマイクロプロセッサは最も性能価格比が良く大型汎用計算機は最悪である。VLSIプロセッサによる大規模アレイ型計算機は，プロセッサ数を多くすることにより，原理的にはいくらでも高速化出来，近い将来，科学計算では本命と考えられている。現在のパイプラインを主とするベクトル計算機では，せいぜい数10倍のスピードアップしか期待出来ない。ベクトル計算の技術は，アレイを構成する単位プロセッサ内の計算技術にとどまるであろう。

プロセッサアレイの実用化のためには，ユーザのもつ問題をうまくアレイ上で並列処理するための，並列アルゴリズムの研究がキーポイントとなる。並列処理への関心をもつアルゴリズム研究者の方々が，これから，ますますふえることを私達は期待している。筑波大のPACS-32システムは，それらの研究者の方々に開放されており，共同研究の申し出を歓迎している。

参考文献

(1). S.F. Lundstrom, et. al.: "A Controllable MIMD Architecture", Proc. 1980 International Conf. on Parallel Processing, IEEE Compt. Soc. (1980).

(2)加藤満佐夫・苗村憲司：並列処理計算機，オーム社(1976).

(3)星野　力他：「科学技術計算専用並列計算機 PACS」7E-8, 7E-9, 情報処理学会第23回 全国大会(1981.10).

(4)星野　力：「並列計算機 PACS-32 による BWR 炉心計算」日本原子力学会誌, Vol. 23, PP. 598～606 (1981).

(5) C.ミード, L.コンウェイ 共著(菅野, 榊訳): 超LSIシステム入門, 培風館 (1981).

(6) R.S. Varga : Matrix Iterative Analysis, Prentice-Hall (1962).

(7)二村良彦・川合敏雄： bit, vol 12, p. 580 (1980).

(8) T. Hoshino, T. Shirakawa, "Load Follow Simulation of Three-Dimensional Boiling Water Reactor Core by PACS-32 Parallel Microprocessor System."

(9)矢嶋信男・野木達夫：発展方程式の数値解析，岩波(1977).

(10)伊藤八大他：「専用並列計算機 PACS によるシミュレーション(II)」昭和56年 日本原子力学会年会 D18, (1981).

(11)星野　力：「PACS による粒子シミュレーションの評価」核融合研究 Vol. 44/別冊3, 名大プラズマ研(1980年).

(12)阿部, 天野： 私信, 情報処理学会(57年3月)発表予定

図1

間接的写像

例 FMP (Flow Model Processor)

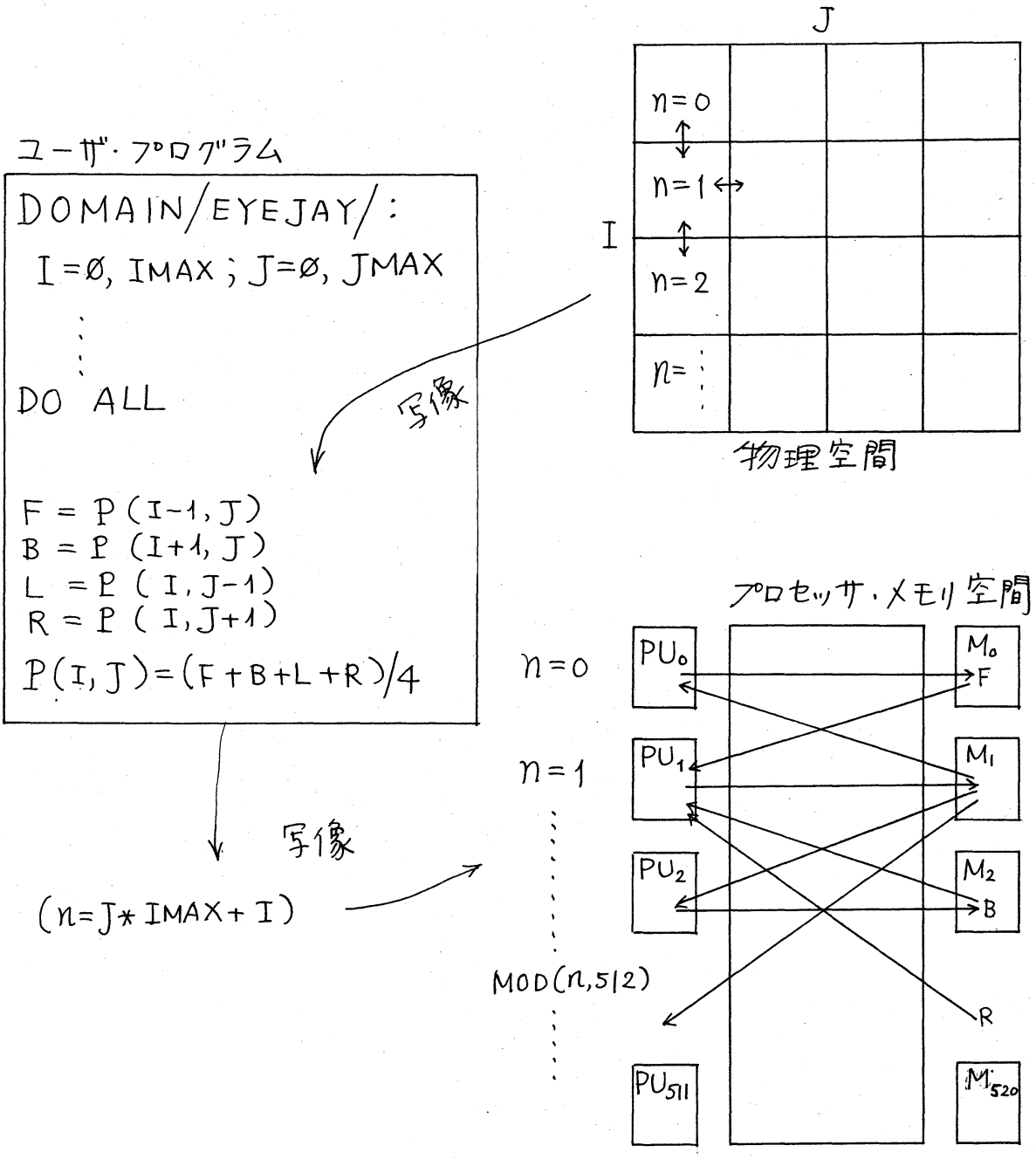
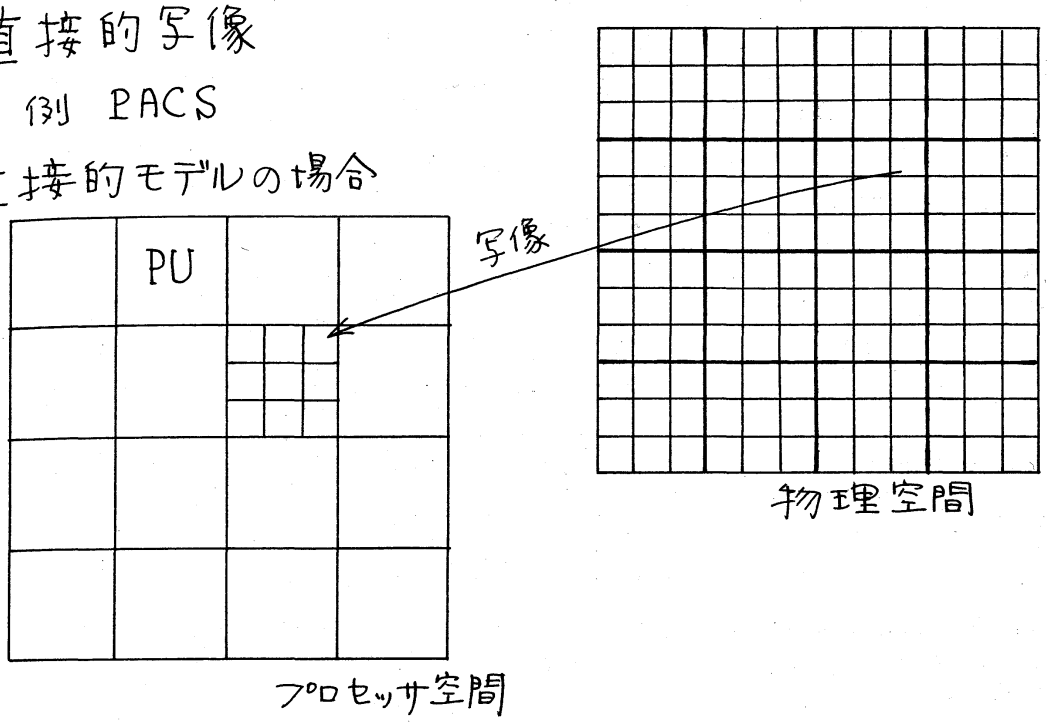


図2

直接的写像

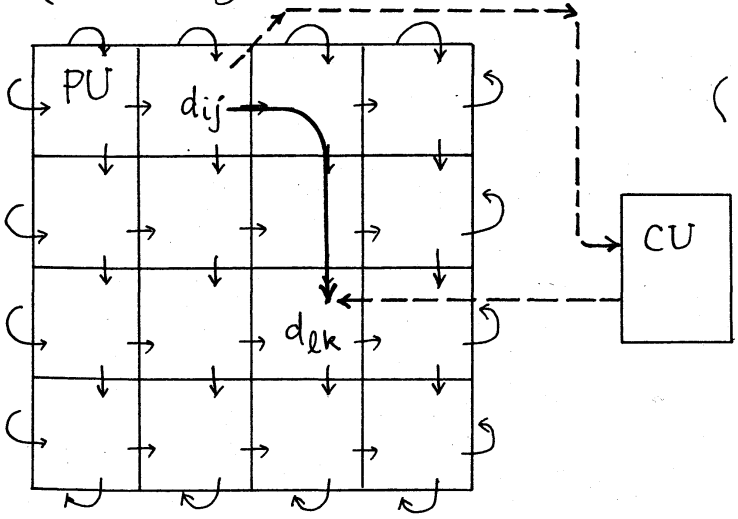
例 PACS

◦近接的モデルの場合



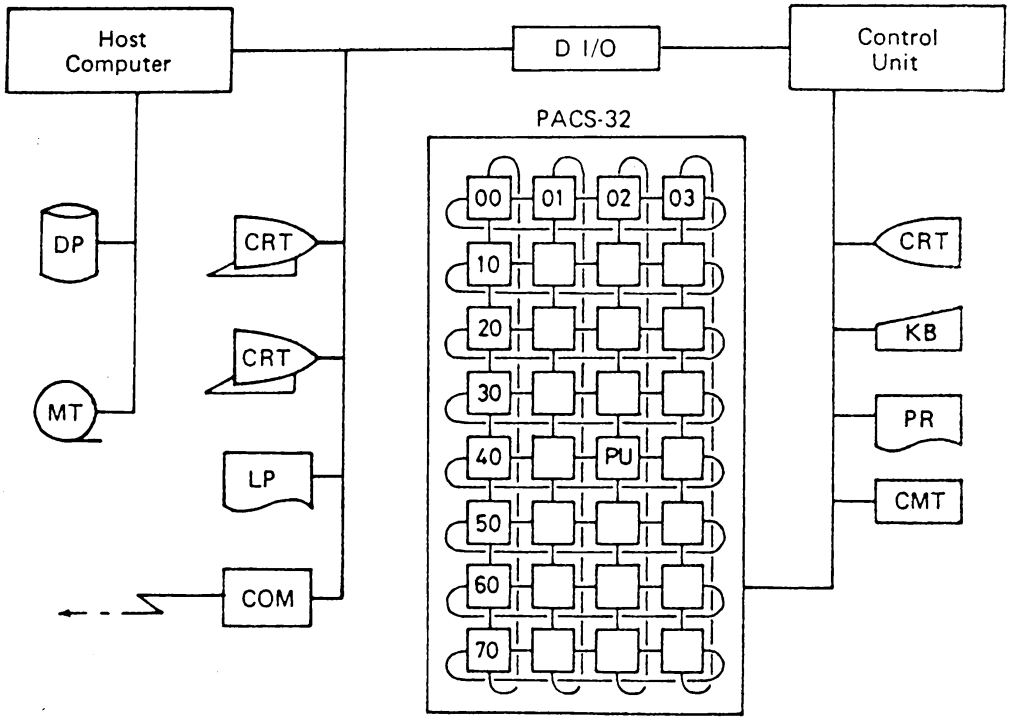
◦非近接的モデル

(Routing または CU 経由通信) + 近接的計算



宛先 データ

(l, k ; d) を Routing する。



Configuration of the PACS system where CRT = cathode ray tube; KB = keyboard; PR = printer; CMT = cassette magnetic tape; DI/O = digital input/output lines; PU = processing unit; LP = line printer; COM = communication interface; DP = disk pack memory; and MT = magnetic tape.

圖 3

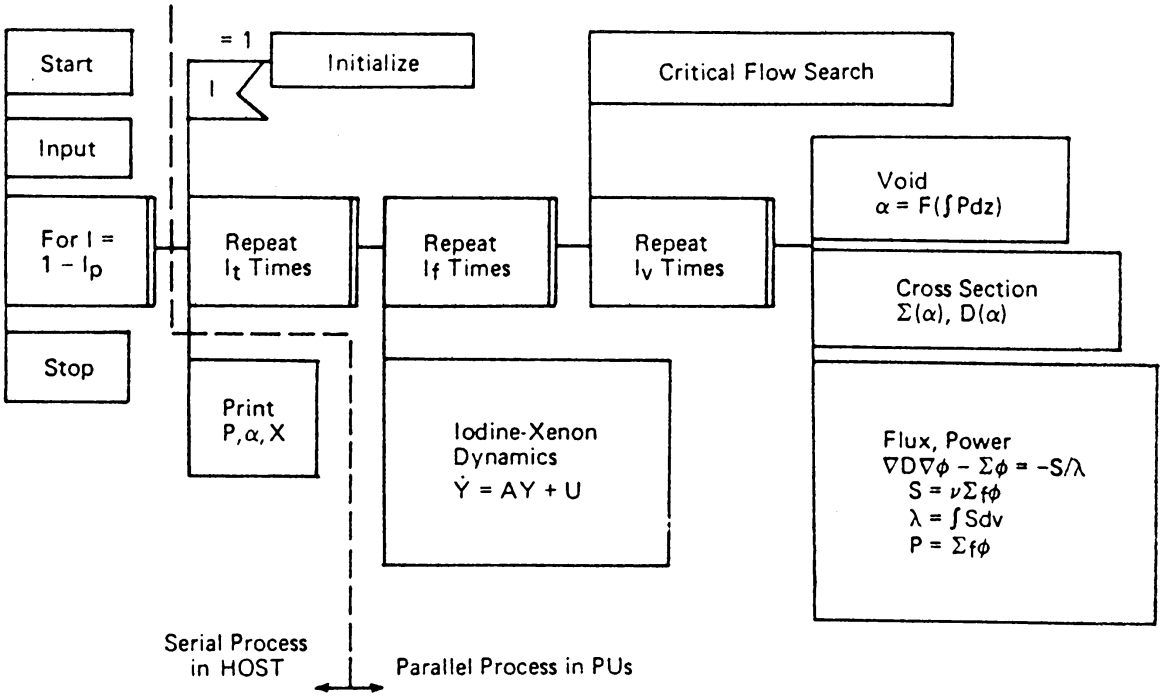
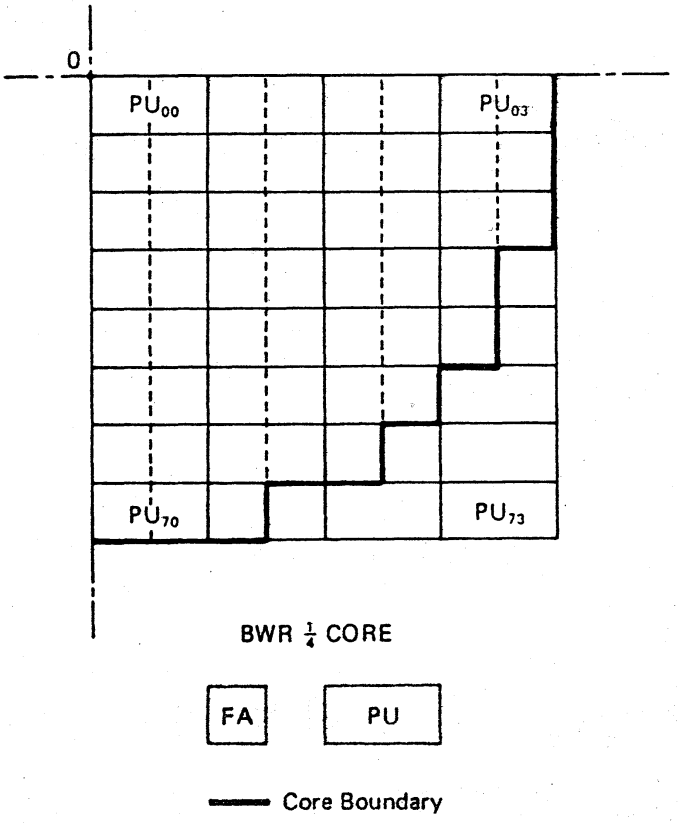


圖 4 The PAD representation of BWR simulation model.

図5



Correspondence of fuel assemblies in a one-quarter symmetric core with PU array.

図6. $\Delta\Phi + f(\Phi)\Phi + S = 0$ の計算.